

LJGG @ CLEF JOKER Task 3: An improved solution joining with dataset from task 1

Leopoldo Jesús Gutiérrez Galeano¹

¹University of Cadiz, 28 Paseo de Carlos III, Cádiz, 11003, Spain

Abstract

In this paper, we describe the results of our participation to the CLEF JOKER 2022 Task 3, "Translate entire phrases containing wordplay". The purpose of this task is the translation of English phrases, which contain wordplay, to French phrases. Our contribution starts explaining the implementation of a basic solution, training just one model, using the given data for task 3. Since we wanted to improve results, we developed a 3-step architecture, which basically is the training of three models: two of them calculate additional information to concatenate to the English phrase, as input for the third neural network. After the generation of results, we decided to translate the whole dataset using DeepL translator, in order to finally compare results between this system and our improved implementation.

Keywords

Wordplay, Pun, Computational Humour, Machine Translation, Neural Networks

1. Introduction

This article contains the strategy for the development of an automatic pun and humour translation system, proposed in the CLEF Workshop JOKER 2022 [1]. They propose three pilot tasks, using the datasets they have prepared for each one, and an unshared task, which accepts any type of submission that uses the provided data. The first pilot task is "classify and explain instances of wordplay", the second one is "translate single words containing wordplay" and the third task is "translate entire phrases containing wordplay".

We have chosen task 3, which basically requires the translation of English phrases, that contain wordplay, to French phrases. Due to a classic neural network trained to generate translations does not usually take care of these kind of senses, we expect better translations using several models instead of just one.

In order to perform this task, we carried out different approaches. Initially, we implemented a basic solution, just using the provided data prepared for task 3. We selected the model mT5-base and we fine-tuned a pre-trained model using the wrapper library SimpleT5. The problem we have seen was training a simple model, in a simple way, without marking any peculiarities, that is, passing just an English phrase as input.

Therefore, after that, we decided to perform a more complex strategy, building an architecture based on three models, through which we could point out characteristic parts for the given

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy


✉ leopoldo.gutierrez@uca.es (L. J. Gutiérrez Galeano)

🌐 <https://www.linkedin.com/in/leopoldogutierrez> (L. J. Gutiérrez Galeano)

🆔 0000-0001-8322-8470 (L. J. Gutiérrez Galeano)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

	id	en	fr
0	pun_11_1	"A good #deed is never lost. Character is #pro...	Une bonne action n'est jamais perdue. C'est la...
1	pun_11_2	"A good #deed is never lost. Character is #pro...	On reconnaît la valeur d'un homme à ses action...
2	pun_11_3	"A good #deed is never lost. Character is #pro...	Quel est le comble pour un notaire? De transme...
3	pun_18_1	"Dad, I'm cold." "Go stand in the corner, I h...	Où envoie-t-on un vilain petit canard? On l'e...
4	pun_18_2	"Dad, I'm cold." "Go stand in the corner, I h...	"Papa, j'ai froid." "Va au coin, il fait 90 d...

Figure 1: The first five rows of the provided dataset for task 3

phrases. So, for the development of this experiment, we decided to enlarge the dataset with the data prepared for the task 1. In this way, at first steps, it is possible to obtain special words from a given English phrase, so that, building an input based on the mentioned words and the initial phrase, we could generate a French translation, given the constructed input.

Finally, we decided to use the online translator DeepL, in order to compare results and then check how good were both systems.

2. Experiments

2.1. Dataset

The provided train data for task 3 contains a set of translated wordplay instances, which have the following fields:

- **id:** The wordplay instance unique identifier.
- **en:** The wordplay text in English.
- **fr:** The wordplay text in French.

There are five wordplay instances in the Figure 1.

2.2. Models

We have selected the T5 model for our experiments, which is known as Text-to-Text Transfer Transformer (T5). It is prepared for a diverse set of tasks and we have to keep in mind that all the tasks are in a "text-to-text" format, so we have to pass text as input and output is text too. This text-to-text model is prepared to perform different tasks, including translation, question answering, and classification [2].

We will use the library SimpleT5 for all the experiments since it is a wrapper which makes easier the use of T5 and mT5 models. This library is built on top of PyTorch-lightning, with transformers. It is just needed to use Pandas to deal with inputs and outputs, and it is possible to do any task, such as, summarisation, translation, question-answering, or any other sequence-to-sequence tasks. This library takes care of import, instantiation, downloading pre-trained models and training [3].

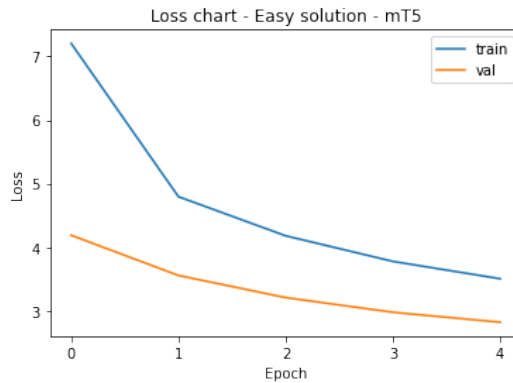


Figure 2: Loss chart for the easy solution

2.3. Hardware and Software Resources

The resources employed for all the experiments are the library SimpleT5, Pandas and sacreBLEU. For the development, we have used Jupyter Notebook. Since t5-large and mt5-base are heavy models, it was needed the use of a high specs machine for the training phase. We used a machine with a GPU NVIDIA Quadro P6000 and 24GB of GPU memory, 30GB of RAM and 8 vCPUs. Anyway, we tried t5-base, t5-small and mt5-small but we decided to select the biggest models for the best machine we could use.

2.4. Previous Experiment

The main objective for the previous experiments is the implementation of a basic solution, based on the training of just one model, which returns French translations, given English phrases. After that, the next goal to reach is the use of this experiment as a reference model, in order to get to know if the model implemented in the main experiment is better.

The main tasks performed to see this experiment through are the following:

- **Data cleaning:** it is possible to find empty values which leads the model to predict wrong values after training. Therefore, the first task to do is the elimination of all missing values.
- **Preparation of dataset:** the selected model expects a dataframe with two columns: “source_text” and “target_text”. We pass the English phrase as “source_text” because this is the input value, and the French phrase as “target_text”, since this is what we want to predict.
- **Training:** for this step we tried the model mT5-base, which initially could be promising since it could return good results. We trained this model with source_max_token_len = 512, target_max_token_len = 128, batch_size = 8 and max_epochs = 5. After training, the best epoch is the fourth and the model accuracy is 0%. This result means that there was not any literal coincidence between translated phrases and expected phrases. The Figure 2 displays the epoch trend.

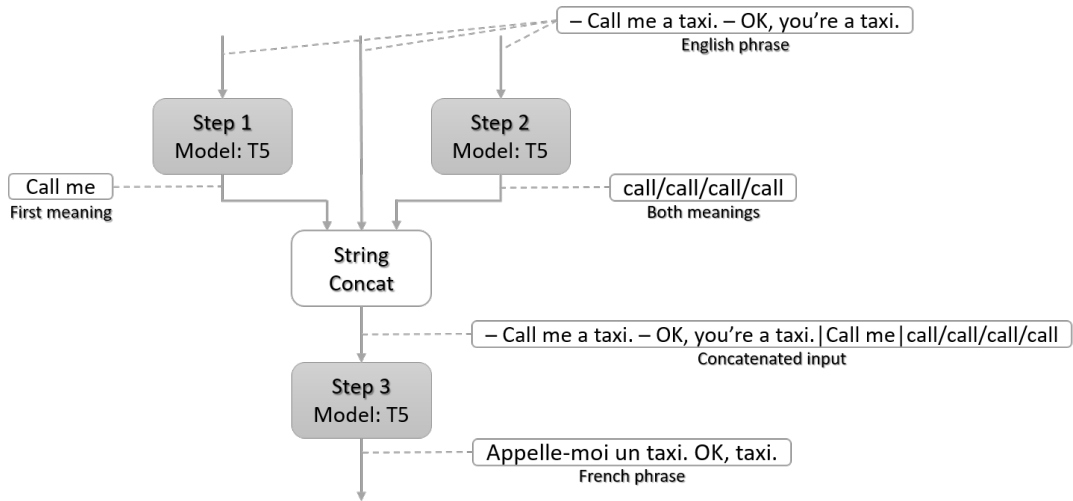


Figure 3: Improved Architecture

2.5. Main Experiment

The main objective of this experiment is the implementation of an architecture, composed by a group of interconnected models, which improve the results of a basic model, like the implemented in the previous experiment. Another goal is testing and comparing different models for relevant parts of the implemented architecture, and finally selecting the best to include in the architecture.

The previous experiment was a fine-tuning of a model with the English phrase as input and the French translation as output. That was a simple task and results could be better if we try to improve the system. In this experiment, we have selected an approach based on an architecture of models, which improve the results. In order to deal with this challenge, we found more data which could be added to the initial dataset provided for task 3. Since task 3 and task 1 datasets have a common field, it was possible to enlarge our dataset with two more fields: one of them is first meaning, which is the first meaning of the pun or wordplay, and the second one is both meanings, which basically is the disambiguation.

The T5 model accepts more than a simple phrase, so we decided to add more information to the model input, in order to get better results. For our improved solution, we use an approach which two or more data are joined used a separator, <sep> [4]. Moreover, there is another approach which uses the symbol | as separator [5]. Therefore, if we pass the English phrase, the first meaning and both meanings, joined together with a separator, we realised that the results were better. In the Figure 3, you can see the architecture with the step 1 model, which predicts first meaning given the English phrase, the step 2 model, which predicts both meanings given the English phrase, and the step 3 model, which returns the French phrase, given the concatenation of the English phrase, a separator, first meaning, another separator and both meanings.

	en	fr	first_meaning	both_meanings
0	- Call me a taxi. - OK, you're a taxi.	Appelle-moi un taxi. OK, taxi.	Call me	call/call/call/call
1	- Call me a taxi. - OK, you're a taxi.	Appelez moi un taxi -- OK vous êtes Un-taxi.	Call me	call/call/call/call
2	- Call me a taxi. - OK, you're a taxi.	- Appelle-moi un taxi - d'accord, tu es un taxi.	Call me	call/call/call/call
3	- Call me a taxi. - OK, you're a taxi.	- Appelez-moi le taxi. - Très bien... TAXIIII...	Call me	call/call/call/call
4	- Call me a taxi. - OK, you're a taxi.	- Appelez moi un taxi. - Très bien, vous êtes ...	Call me	call/call/call/call

Figure 4: The first five rows of task 3 data with two fields added from task 1 data

After studying the provided data for all the tasks, we found some kind of relation between the datasets prepared for task 3 and task 1. The main purpose of improving the experiment could be fulfilled if we join both datasets, just if we attach additional information to the model input, in order to obtain better predictions.

The main tasks performed for this experiment are the following:

- **Data cleaning:** we start with the same data cleaning task performed for the previous experiments. Besides that, since we are going to join data provided for task 1, we need to clean the same kind of missing values for that dataset.
- **Joining task 3 and task 1 data:** the dataset prepared for task 1 contains a bunch of fields, which three of them are useful for our experiment. The “WORDPLAY” field, in task 1 data, contains the English phrase, that is, the “en” field in task 3 data. This fact makes possible joining more fields to the initial dataset. We will add the fields “TARGET_WORD” and “DISSAMBIGUATION”, which will be renamed to “first_meaning” and “both_meanings”, respectively. A sample of data, after enlarging the initial dataset with two fields more from task 1 data can be seen in Figure 4.
- **Looking for separators:** since the main goal of this experiment is improving results, and we will reach this purpose by passing the new fields, “first_meaning” and “both_meanings”, to the model input, we need a way to pass a different input to the model. Due to the model expects a string as input, we need to build a new string with the English phrase and both fields. In order to solve this issue, we have to find a proper separator to avoid ambiguities. For instance, in the Figure 4, you can see that the “both_meanings” field contains the slash character, therefore we cannot use it as separator. We studied all the fields we are using and the hash character is used, as well as, the dollar, the “at” symbol or the ampersand. Finally, we found that the vertical bar was not used in the dataset, so we have selected that character. In the Figure 5, you can find an outline of how the new concatenated input is formed.
- **Preparing dataset for the three step strategy:** our improved 3-step architecture is composed by three models, which need to be trained. In order to do that, and now that we have cleaned the data, we have to prepare the three datasets needed for this task. Since the first step model will predict the first meaning, given an English phrase, we will create a dataset with two columns: “en” renamed as “source_text” and “first_meaning”

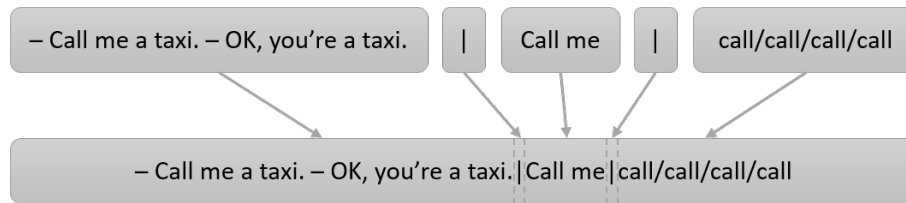


Figure 5: The new concatenated input for the improved architecture

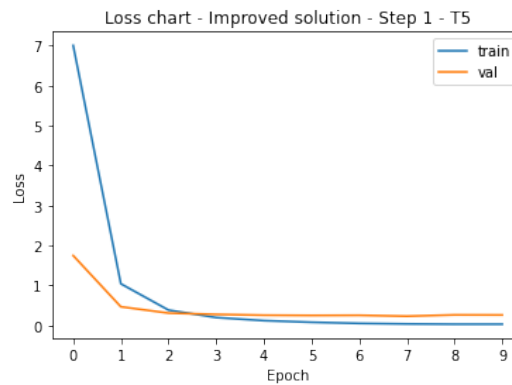


Figure 6: Loss chart for the improved solution - Step 1

renamed as “target_text”. Similarly, the second step model will predict both meanings, given an English phrase, so we will create a dataset with two columns: “en” renamed as “source_text” and “both_meanings” renamed as “target_text”. Finally, since the third step model will predict the French phrase, given a concatenation of the English phrase, first meaning and both meanings, we will create a dataset with two columns: the concatenated field named as “source_text” and “fr” renamed as “target_text”.

- **Training the step 1 model:** for this step, we used the model t5-large, with the following parameters: source_max_token_len = 50, target_max_token_len = 50, batch_size = 16 and max_epochs = 10. After training, the best epoch is the seventh and the model accuracy is 79,36%. The Figure 6 contains a chart, which shows train and validation loss with respect to each epoch.
- **Training the step 2 model:** we used the same model and parameters used for the step 1 model. In this case, the best epoch is the third and the model accuracy is 19,04%. We have to keep in mind that the accuracy was calculated by comparing if strings are the same, that is, we did not study the real accuracy by studying if each result was actually equivalent or valid. The Figure 7 contains a chart, which shows train and validation loss with respect to each epoch.
- **Training the step 3 model with mt5-base:** we tried a different model, mT5-base, which initially could be promising since it could return better results. We trained this model with source_max_token_len = 512, target_max_token_len = 128, batch_size = 8 and max_epochs = 5. After training, the best epoch is the fourth and the model accuracy

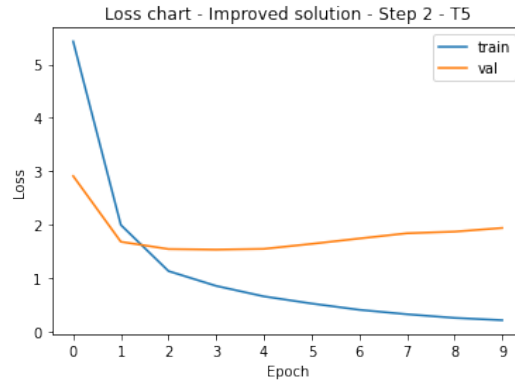


Figure 7: Loss chart for the improved solution - Step 2

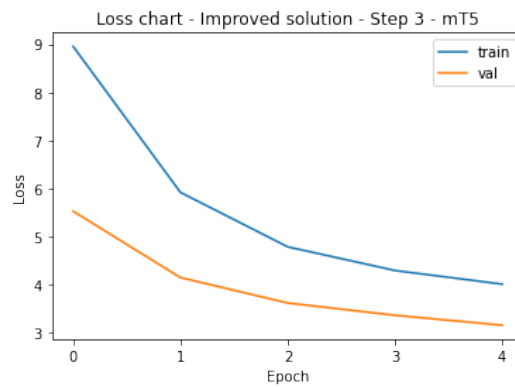


Figure 8: Loss chart for the improved solution - Step 3 with mT5

is 0%. This result means that there was not any literal coincidence between translated phrases and expected phrases. The Figure 8 contains a chart, which shows train and validation loss with respect to each epoch.

- **Training the step 3 model with t5-large:** in this case, we used the same model and parameters used for steps 1 and 2 models. In this case, after training, the best epoch is the second and the model accuracy is 0,17%. We have to keep in mind the same difficulties to get a realistic accuracy than we had for the previous models, due to the comparison of literal strings. The problem is that we are expecting a concrete translation and the model returns a translation, which could be valid, but we count it as invalid. The Figure 9 contains a chart, which shows train and validation loss with respect to each epoch.

2.6. Analysis of results

We analysed results using the BLEU score. For the previous experiment, we obtained 4.88 using the mT5 model. This means that translations generated are almost useless. For the improved solution, in step 3, we obtained 3.14 for the mT5 model, which means that translations are

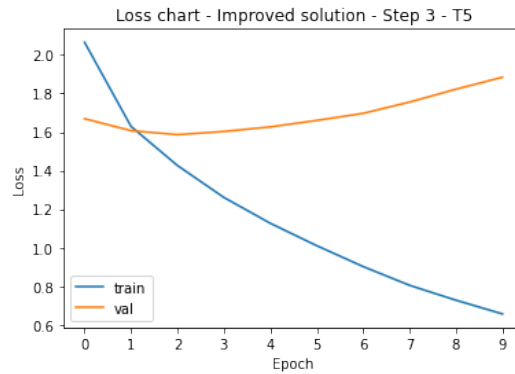


Figure 9: Loss chart for the improved solution - Step 3 with T5

almost useless too. However, for the T5 model we got 19.99, which means that it's hard to get the gist. We have to say that the obtained score is almost 20, so the feedback could be between that one and the next one, for scores between 20 and 29, which says that the gist is clear but has significant errors [6]. Moreover, phrases are in French. Anyway, the third step is better with t5-large, since we got better BLEU score and better accuracy. Therefore, we have selected the improved architecture, with t5-large for step 3, as the best implementation.

The results obtained were sent to the organization and were ranked as the third best submission, out of seven. All the teams sent a total 2378 translations. We have seen that this is the third best submission with more valid translations, 2264, in contrast to 206 not translated phrases and 349 with non sense. This submission generated more untranslated and non sense phrases than the others. However, 46 phrases have syntax problems and 78 have lexical problems, and just one submission was worse for both indicators. 1595 translations preserve the lexical field of the source wordplay, 1327 preserve the sens of the source wordplay and 827 uses comprehensible terms, which are the fourth best indicators. 261 corresponds to translations that are wordplay, 240 are wordplays that are understandable for general audience, 4 have style shift, e.g. in case whether a vulgarism is present either in a source wordplay or in a translation but not in both and 838 hilariousness shift, or translations that were judged much or much less funny than the source wordplay, which are in the third position. And we sent 9 over-translations, or translations that have useless multiple wordplay instances when the source text has just one, which is the worst value [7].

3. Manual translations with DeepL

DeepL is an online translator, which could be used for free with limitations or with a subscription if we want to use more features. Their translations are direct modifications of the Transformer and the neural networks of DeepL also contain different parts of this architecture, such as attention mechanisms [8].

We submitted DeepL as manual translations since we had to copy and paste English phrases and French translations in batches of 70 or 80 phrases, due to the limitation of characters for

the free version.

The main objective was getting good results and comparing them with the automatic translations returned by the model implemented in the main experiments.

This results were sent to the organization and were ranked as the best submission. 2378 translations were sent and 2324 were valid, so this submission is the second in valid translations. 39 phrases were not translated, 59 have non sense, 17 have syntax problems, 25 have with lexical problems and 3 over-translations, and all these indicators obtained the third best values. The following indicators were the best: 2184 translations preserve lexical field of the source wordplay, 1938 preserve the sense of the source wordplay, 1188 use comprehensible terms, 373 are translations that are wordplay and 342 have identifiable wordplays. Finally, these submission indicators were ranked as the second best: 9 have style shift and 930 hilariousness shift [7].

To sum up, DeepL results were better than the improved solution submission for all indicators. Moreover, it was better than the second classified for all indicators, except for two: over-translations and style shift [7].

4. Conclusions and Perspectives for future work

Results obtained using DeepL thrown the best results. They have a translation service which has been improved over the years. In spite of that, the results generated by the developed experiments are showing that the best results were obtained using an improved architecture of models. Although the automatic results were ranked in third position, this solution could be refined in the future.

The main perspective for future work is trying to improve the results, changing the implemented architecture in the main experiment. For step 3, the input was English phrase, separator, first meaning, separator and both meanings. Another idea could be checking results if we swap the information in the concatenation step. For instance, it could be possible to get better translations if we pass first meaning, separator, both meanings, separator and English phrase. Another additional way could be by adding more data from task 1 dataset, converting to lowercase all characters or maybe separating step 3 in two models, one for puns and the other one for wordplays.

Additionally, another model could improve results. For instance, mt5-large, or maybe any other. Trying different parameters for the model or using T5 directly with PyTorch, instead of SimpleT5, could be another way to get better translations.

References

- [1] L. Ermakova, T. Miller, O. Puchalski, F. Regattin, É. Mathurin, S. Araújo, A.-G. Bossler, C. Borg, M. Bokinić, G. L. Corre, B. Jeanjean, R. Hannachi, Ğ. Mallia, G. Matas, M. Saki, CLEF Workshop JOKER: Automatic Wordplay and Humour Translation, in: M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvgå, V. Setty (Eds.), *Advances in Information Retrieval*, volume 13186, Springer International Publishing, 2022, pp. 355–363. doi:10.1007/978-3-030-99739-7_45.

- [2] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. URL: <https://arxiv.org/abs/1910.10683>. doi:10.48550/ARXIV.1910.10683.
- [3] R. Shivanand, SIMPLET5 - train T5 models in just 3 lines of code, 2021. URL: <https://medium.com/geekculture/simplet5-train-t5-models-in-just-3-lines-of-code-by-shivanand-roy-2021-354df5ae46ba>.
- [4] D. Mullick, A. Fyshe, B. Ghanem, Discriminative models can still outperform generative models in aspect based sentiment analysis, 2022. URL: <https://arxiv.org/abs/2206.02892>. doi:10.48550/ARXIV.2206.02892.
- [5] W. Zhang, X. Li, Y. Deng, L. Bing, W. Lam, Towards generative aspect-based sentiment analysis, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics, Online, 2021, pp. 504–510. URL: <https://aclanthology.org/2021.acl-short.64>. doi:10.18653/v1/2021.acl-short.64.
- [6] Google, Evaluating models, 2022. URL: <https://cloud.google.com/translate/automl/docs/evaluate>.
- [7] L. Ermakova, T. Miller, F. Regattin, A.-G. Bosser, É. Mathurin, G. L. Corre, S. Araújo, J. Boccou, A. Digue, A. Damoy, P. Campen, B. Jeanjean, Overview of JOKER@CLEF 2022: Automatic Wordplay and Humour Translation workshop, in: A. Barrón-Cedeño, G. Da San Martino, M. Degli Esposti, F. Sebastiani, C. Macdonald, G. Pasi, A. Hanbury, M. Potthast, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), 2022, p. 25.
- [8] DeepL, How does DeepL work?, 2022. URL: <https://www.deepl.com/sv/blog/how-does-deepl-work>.